LANGUAGE IN INDIA

Strength for Today and Bright Hope for Tomorrow

Volume 6 : 10 October 2006

Managing Editor: M. S. Thirumalai, Ph.D.
Editors: B. Mallikarjun, Ph.D.
Sam Mohanlal, Ph.D.
B. A. Sharada, Ph.D.
A. R. Fatihi, Ph.D.
Lakhan Gusain, Ph.D.
K. Karunakaran, Ph.D.
Jennifer Marie Bayer, Ph.D.

A SURVEY OF THE STATE OF THE ART IN TAMIL LANGUAGE TECHNOLOGY

S. Rajendran, Ph.D.

A SURVEY OF THE STATE OF THE ART IN TAMIL LANGUAGE TECHNOLOGY

S. Rajendran, Ph.D.

A PRELUDE

The use of computer for language analysis leads to the technological development of languages in general, and Tamil, in particular. The world scenario has its impact on Tamil language too. Both the government and private organizations have initiated programs for the technological development of Tamil language. The Department of Electronics had conducted training Courses on Natural Language Processing through selected institutions throughout India and paved way to technological development of Tamil. It funded Machine Translation programs among Indian languages and between English and Indian languages. It also funded for the development of corpus for Indian languages. It had identified certain centres for the Technological Development of Indian languages and funded them to initiate projects, which aims to achieve their goal.

Anna University at Chennai had been identified for the technological development of Tamil language and provided with a fund of a few crores of rupees to fulfill this mission. Under this scheme a Resource Centre for Indian Language Technology Solutions-Tamil has been established at Anna University. A team of researchers employed under the scheme has prepared a number of Language Technology Products. This has lead to the technological development of Tamil in many areas. Many other organizations, both government and private, followed this.

Tamil University at Thanjavur, Tamil Virtual University, AUKBC Research Centre at Chennai, Central Institute of Indian Languages at Mysore and International Forum for Information Technology in Tamil (INFITT), which conducts international conference of Tamil internet every year, put their efforts for the technological development of Tamil. Apart from the above institutions IIT, Chennai, IISC, Bangalore, and Micro Software, Bangalore also have contributed for the technological development of Tamil.

In this paper the technological development of Tamil has been classified under certain heads and the research works under taken and successfully completed as well as the products made are discussed in details.

1. CORPUS AND CORPUS MANAGEMENT TOOLS

Corpus linguistics seeks to further our understanding of language through the analysis of large quantities of naturally occurring data. There is a long tradition of corpus linguistic studies in Europe. The need for corpus for a language is multifarious. Starting from the preparation of a dictionary or lexicon to machine translation, corpus has become an inevitable resource for technological development of languages. Corpus means a body of huge text incorporating various types of textual materials, including newspaper, weeklies, fictions, scientific writings, literary writings, and so on. Corpus represents all the styles of a language. Corpus must be very huge in size as it is going to be used for many language applications such as preparation of lexicons of different sizes, purposes and types, machine translation programs and so on.

1.1. Tagged corpus, Parallel Corpus, and Aligned Corpus

Corpuses can be distinguished as tagged corpus, parallel corpus and aligned corpus. The tagged corpus is that which is tagged for part-of-speech. A parallel corpus contains texts and translations in each of the languages involved in it. It allows wider scopes for double-checking of the translation equivalents. Aligned corpus is a kind of bilingual corpus where text samples of one language and their translations into other language are aligned, sentence by sentence, phrase by phrase, word by word, or even character by character.

CIIL Corpus for Tamil

As for as building corpus for the Indian languages is concerned it was Central Institute of Indian languages (CIIL) which took initiative and started preparing corpus for some of the Indian languages (Tamil, Telugu, Kannada, and Malayalam). Department of electronics (DOE) financed the corpus-building project. The target was to prepare corpus with ten million words for each language. But due to financial crunch and time restriction it ends up with three million words for each language. Tamil corpus with three million words is built by CIIL in this way. It is a partially tagged corpus. This corpus is available in CD and one can get a free copy from CIIL for research purpose. At present CIIL is planning to build corpus with 10 million words for Indian languages.

AUKBCRC's Improved Tagged Corpus for Tamil

AUKBC Research Centre which has taken up NLP oriented works for Tamil, has improved upon the CIIL Tamil Corpus and tagged it for their MT programs. It also developed parallel corpora for English-Tamil to promote its goal of preparing

an MT tool for English-Tamil translation. Parallel corpus is very useful for training the corpus and for building example based machine translation. Parallel corpus is a useful tool for MT programs.

1.2. Corpus Indexing Tools (Concordance, KWIC index, etc.)

Many such tools have been made for Tamil. Few important ones are listed below:

CIIL Corpus Indexing Tool

CIIL has prepared a corpus-indexing tool for indexing Tamil corpus prepared by the institute.

Tamil University's Indexing of Sangam Literature

Department of Computer Science and Department of Lexicography of Tamil University, Thanjavur have prepared a tool for finding concordance and indexing Tamil Sangam literary works. They have papered an Index of Tamil Sangam Literature using computer. This is going to be printed by Tamil University.

AUKBCRC Corpus Indexing Tool

Again AUKBCRC has to be quoted for preparing a tool for corpus indexing. The tool can sort, alphabetize and give concordance and KWIC index for Tamil texts in any font. It can find the frequency of lexical items, both lemmatized and inflected forms.

Ganesan's Tool for Corpus Analysis

Ganesan and Rajan are involved in making tools for corpus analysis.

1.3. Corpus Compression and encrypton tools

Corpus needs to be compressed for certain purpose. Also important data and messages need to be encrypted to send them through Internet without being intercepted. There are research works in this line of thinking for Tamil. The following works come to my knowledge:

Baskaran's study on Zip's Law for Tamil

Baskaran of Tamil University has studied the possibilities of using Zip's law for the compression of Tamil texts.

AUKBCRC's Research on Encryption

AUKBC Research Centre is involved in extensive research on encryption. Encryption tools help in exchanging information from on end to another end without being intercepted, sabotaged or spied. There are attempts for preparing such tools for Tamil too.

1.4. Text Processing Tools

Much of the information processed by computers is texts, data of the type generally called character or alphanumeric. In natural language processing, all written material is text. Therefore, text processing and the input and output of text are important.

Dealing with text is both simpler and harder than manipulating numeric data.

In terms of the physical characteristics, it is simpler in that text is linear; the first character is handled, then the second, then the third, until the last is reached. At that point the data is processed. Both in logical terms, text is quite slippery. Generally, in the computer, numeric data is represented in a specific form: a number is given a fixed quantity of bits and a set format. All integers occupy the same amount of storage in memory in a particular computer, as do real numbers. Text, on the other hand, is made up of words and names and other strings of characters, which are many different lengths, and thus require differing amounts of memory storage.

AUKBCRC's Text Processing tool

AUKBC Research Centre has prepared a tool for Tamil text processing. It is a reasonably good tool for all sorts of string analysis. Using the morphological analyzer and syntactic parser papered by the NLP team of AUKBCRC, the processor is able to give all sorts of linguistics information one can expect form text analysis. It can tag the text with tab labels, separate the text into sentences, phrases, and words, count the frequency of occurrence of words in terms of their tokens and word forms. It can count the number of characters of a specified length. All sorts of statistical information on string analysis can be had form the text processor.

1.5. Statistical Analysis Tool

Many statistical analysis tools have been made for Tamil. Both Anna University RCILTS-T and AUKBCRC have made statistical analysis tool for Tamil. The following work comes to my knowledge:

AUKBCRC's Statistical Analysis Tool

AUKBC Research Centre at Cennai has prepared a tool which works on Tamil corpus and gives statistical information in terms of number of word forms, number of words (lexemes) or word tokens and number of lines and paragraph in a text. The tool is an extremely useful one.

2. TEXT EDITORS AND WORD PROCESSOR

Many text editors and word processors are developed for Tamil.

2.1. Text Editing Tools

Text editing tools are available for Tamil.

2.2. Word Processing Tools

Word processing refers to the activity carried out using a computer and a suitable software to create, view, edit, manipulate, transmit, store retrieve and print documents. A document may contain text, tables, graphs, equations, pictures and drawings. Word processor is used to produce documents for business or personal use such as newsletter, reports, letters and essays. One might say that a word processor is an intelligent typewriter. One can type a whole page, make corrections (editing), use more than one typefont to give beauty to the text, checkup paragraphs in different styles and columns (formatting) and also check spelling, find and insert synonyms for a word (thesaurus), and process it in many more ways before we actually put that page to pint. Generally all the word processing efforts can be listed as follows:

- Creating a New Document
- Entering Text in a Document
- Saving, Closing and Opening Documents
- Moving around the Document
- Scrolling the Document
- Correcting Mistakes
- Inserting Text
- Moving the Text
- Copying the Text
- Searching and Replacing the Text

A few of Word Processing tools, which are the outcome of extensive research on Tamil word processing, are available for Tamil. Many commercial word processors are also available for Tamil. A few of them are listed below:

CIIL Word Processor for Tamil

Many word processors are available in the market for Tamil. CIIL, Mysore prepared a word processor for Tamil named Bharati which was available in CD form.

ILipi: C–DAC of Pune prepared a word processor for Indian Languages named ILipi which supports Tamil too.

Word Processor of Deyvasundaram

Under the supervision of Professor Deyvasundaram of Madras Universit,y Switha computers, Chennai has prepared a word processor for Tamil with tools like spell checker, morphological analyzer, etc.

Ilango Tamil Word Processor has the following features: highlights all incorrect words, add new words to dictionary, single word and full story checking, and check for sandhi errors.

Shakthi English/Tamil Office Suite (prepared by Chnneai Kavigal, Kanini Pvt. Ltd, Chennai) has Tamil Word Processor named *patami.*

Palagai: The Resource Centre for Indian Language Technology Solutions – Tamil functioning (RCILTST) at the School of Computer and Engineering, Anna University, Chennai 25, Tamilnadu, India has prepared a Tamil Word Processor named palagai. Palagai provides basic facilities for word processing both in Tamil and English. It supports a spell checker, a grammar checker and an e-mail facility. It has the following salient features:

- Files of both Rich Text and Text-only formats can be created and edited.
- HTML files can be viewed
- E-mails can be sent
- Provisions to check the spelling and grammar
- Supports Sort, Find and Replace operations
- Print facility is provided
- Unlimited Undo and Redo operations

Navaneethan's Multi Word Processor with online Noun Translator

Navaneethan and his team (Navaneethan et al, 2004) have prepared a multi word processor with online noun translator. This word processor has been built on top of PANDITHAM (Protocol for Applications Development in Thamizh and Multilingual computing), protocol, which provides an efficient framework for taking multiple languages into the machine. Any document opened using this processor can invoke the translator facility by selecting a Noun and using an appropriate

accelerator. Other features of the processor include multilingual support, switching between environment and application language.

2.3. DTP tools

Desktop publishing leaves the doors to writing and publishing wide open. We are not locked out from the tools we need. We can use our personal computer to print and publish whatever is important to us, whatever we believe people should know. This makes desktop publishing a very existing endeavour.

Desktop publishing is an alternative to professional publishing. With a personal computer, page layout software, and a printer, we can perform most of publishing a document ourselves. Our personal computer allows us to write and edit text with ease. Page layout software enables us to design and compose design and compose pages, integrating text and graphics. Laser printers print such good-looking text that we can avoid the expense of typesetting documents. With a little publishing experience, we can produce effective, well-produced publications less expensively than commercial publisher can.

Desktop publishing is largely a do-it-yourself operation. We the publisher are responsible for a wide variety of task. We will do almost all of the planning, managing, writing, editing, and production work that go into our publication. We may have one or perhaps two other people working with us, but a larger staff will take you out of desktop publishing as defined in the literature. People who are no publishers do desktop publishing, people with little, if any, publishing experience.

As far as Tamil is concerned, many DTP software are available. DTP printing has replaced traditional printing techniques, which are very costly and time consuming. Due to the growth and popularity of DTP, more number of Tamil publications are now brought out. The publication of books and journal in Tamil has drastically increased due DTP facility.

2.4. Fonts

Innumerable fonts are available for Tamil. The availability of too many fonts makes text recognition difficult as each font makes use of different codes. It nullifies uniformity. The multiplicity of fonts makes the computational processes in Tamil difficult. A few years back, Tamilnadu government intervened and proposed standard fonts using standard codes which led to the development of TAM and TAB fonts. There are plenty of fonts for Tamil such as Diamond, Kampan, Tiruvalluvar, Tolkappiyan, Kannaki, and so on.

3. DICTIONARY TOOLS

Tamil has many dictionary tools. Some of the paper dictionaries have been converted into electronic dictionaries. Some are specially made as electronic dictionaries. There are attempts for online dictionaries too.

3.1.Word List/Vocabulary

Sollooviyam

It is one of the content creations of RCILTST, Chennai. *Sollooviyam* is a picture dictionary for children to teach the alphabets and simple words of Tamil language. It has about three hundred words –organized as nouns and verbs. The words are grouped alphabetically under vowels and consonants. Each word has an associated picture, an explanation and English equivalent. Every word has a small poem to illustrate its meaning. This makes the child to imagine the poem and in effect the word gets etched in his/her memory.

Union Catalog of Tamil Palmleaf Manuscript

This is prepared by Tamil University and is available in electronic form. A database is created for cataloging Tamil Palmleaves in Tamilnadu. This project is funded by the Ministry of Education and Culture of the Government of India and undertaken by Computer Centre, Department of Palm Leaf Manuscripts and Library of Tamil University. Information is collected from 49 institutions. The Catalogue consists of two parts viz. Part-A and Part-B. The Part-A consists of entries arranged in a classified sequence of subjects. In Part-B, [1] Author Index [2] Commentator Index [3] Subject wise Index [4] Title Index are provided. In addition, some statistical details of the collection are given under the heading bibliometry. The total number of entries available in the catalogue is 21, 973.

Scientific Data Base for technical terms through computer: This is prepared by Department of Computer Science, Tamil University, Thanjavur. As an experiment measure a system has been developed to create a scientific data base for the basic science subjects such as mathematics, physics, and chemistry to help a user to write books and articles in Tamil. The system provides an equivalent technical term in Tamil for a given word or group of words in English and vice versa.

3.2. Electronic/Online dictionaries

Many electronic/online dictionaries have been prepared for Tamil by research institutions as well as commercial organizations. A few of them are listed below:

Radha Chellappan's Electronic dictionary for Scientific Technical Terms in Tamil

It a very exhaustive and efficient electronic dictionary of scientific terms in Tamil. It provides us with synonymous words along with the standardized lexical items. It has different kinds of retrieval and browsing facility. Her third version is much

more efficient and more informative than her earlier versions. One can straight away get the dictionary in printed form too.

coRputaiyal: It is also one of the content creations of RCILTST, Chennai. *Sorputhaiyal* is a language oriented software for the retrieval of lexical entries from monolingual and bilingual dictionaries by many users simultaneously. This Tamil Online Dictionary contains 20,000 root words. Each entry in the dictionary includes the Tamil root word, its English equivalent, different meaning of the word, and the associated syntactic category. The root words are classified into 15 categories. It supports a user-friendly interface to provide the information. The tool uses the morphological analyzer, which retrieves the root word form the word given by the user. Hence even if the user specifies an inflected word, the dictionary fetches the root word and gives all relevant information about the word.

Pals e-Dictionary: It is an English-English-Tamil dictionary. It is available in CD-ROM. It has 22,000 headwords and 35,000 sub-words. The supporting software enables one to turn the pages easily. And also the search can be made quickly. Pals e-dictionary is available for browsing in Tamil Virtual University website.

Tamil Lexicon of Madras University: Tamil Lexicon published by Madras University is available in e-form in Tamil Virtual University Website.

English-Tamil Dictionary of Madras University: English-Tamil dictionary published by Madras University is also available in e-from in Tamil Virtual University Website.

Tamil-Tamil Dictionary of M.Shanmugam Pillai: Tamil-Tamil dictionary prepared by Shanmugam Pillai is now available in e-form in Tamil Virtual University Website.

Multidimensional SMART dictionary: Vijaya and John Paul (2004) proposed an electronic dictionary for Tamil named 'Multidimensional SMART dictionary', a project work of CIIL. The Multidimensional SMAT (Small Readable Tamil) is an advanced Tamil dictionary, which has the following unique features: it is a web based dictionary and it is open to all at any time. The dictionary will give some or all of the following details, if anybody searches for a word:

- Search word in Tamil
- Spoken pronunciation/Alternative pronunciation in roman transcription
- Most frequent meanings in English
- Grammatical Categories in English
- Combination / Collocation construction in Tamil, its roman transcription and English meaning
- Compound constructions in Tamil, its roman transcription and English meaning

- Proverbial constructions in Tamil, its roman transcription and English meaning
- Synonyms in Tamil and its roman transcription
- Antonyms in Tamil and its roman transcription

They have collected and entered 6992 head words (search words), their grammatical categories, meanings, synonyms and antonyms. Now

Tamil WordNet (TWN): TWN has been built by Department of Linguistics, Tamil University, in collaboration with AUKBC Research Centre, Chennai. Tamil Virtual University has sanctioned four lakhs of rupees for the project. Rajendran of Tamil University is the Chief Investigator and Arulmozi of AUKBC is the co-investigator. TWN is based on the architecture of EuroWordNet, which is an online lexical database.

Until recently only dictionaries in printed book format represented the lexicon of a language. TWN is a semantic dictionary that is designed as a network, partly because representing words and concepts as an interrelated system seems to be consistent with evidence for the way speakers organize their mental lexicons. TWN design resembles that of a thesaurus in that its building block is a synset consisting of all the words that express a given concept. Thus, the user of a TWN who has a given concept in mind can find, by calling up one of the words expressing this concept, other words that lexicalize the same concept. But TWN does much more than lists concepts in the form of synsets.

The synsets are linked by means of a number of relations, including hyponymy, meronymy, and entailment. Different kinds of semantic oppositions lumped together in the antonymy relation link words only, rather than concepts. TWN thus clearly separates the conceptual and the lexical levels, and this distinction is reflected in the one between semantic-conceptual and lexical relations that hold among synsets and words, respectively.

Unlike thesaurus, the relations between concepts and words in TWN are made explicit and labeled; users select the relation that guides them from one concept to the next and choose the direction of their navigation in conceptual space. Words express concepts, and the lexicon is constrained by the kinds of concepts that are available to us by virtue of our perception of, and interaction with, the world around us. TW differs from thesauruses, where only lexicalized concepts are accounted for. In some respects, TWN resembles a traditional dictionary. For example, TWN gives definitions and sample sentences for most of its synsets.

TWN also contains information about morphologically related words. TWN's goals differ little from those of a good standard college-level dictionary, and the semantics of TWN is based on the notion of sense that lexicographers have traditionally used in writing dictionaries. It is in the organization of that

information that TWN aspires to innovation. TWN does not give pronunciation, derivation morphology, etymology, usage notes, or pictorial illustrations. TWN does however, try to make the semantic relation between word senses more explicit and easier to use.

TWN relies on extensive preliminary investigations of the vocabulary of Tamil (Rajendran, 1976-2003) based on the componential analysis of meaning (Nida, 1975a & 1975b) and structural semantics (Lyons, 1977). Portions of this work have been compiled into a Tamil Thesaurus (Rajendran, 2001).

The Tamil thesaurus in electronic form represents the ontological structure of Tamil (shortly OST) vocabulary giving scope to take care of any kind of semantic/lexical relations that hold between lexical items.

TWN makes the commonly accepted distinction between conceptual-semantic relations, which link concepts, and lexical relations, which link individual words. The mental lexicon tends to build semantic networks with conceptual-semantic relations, whereas workers focusing on lexical aspects use primarily lexical, word-word relations. Semantic relations organize TWN.

Since a semantic relation is a relation between meanings, and since meanings can be represented by synsets, it is natural to think of semantic relations as pointers between synsets. It is characteristic of semantic relations as pointers between synsets. TWN does not contain syntagmatic relations linking words form different syntactic categories. The four major syntactic categories (Noun, Verb, Adjective, Adverb) are treated separately. Nouns are organized in lexical memory as topical hierarchies, verbs are organized by a variety of entailment relations, and adjectives and adverbs are organized as N-dimensional hyperspaces.

Table of lexical/semantic relations for nouns:

Relations	**Subtypes**	**Example**
Synonymy		*puttakam* 'book' to *nduul* 'book'
Hypernymy-Hyponymy		*vilangku* 'animal' to *paaluuTTi* 'mammal'
Hyponymy-Hypernymy		*pacu* 'cow' to *paaluuTTi* 'mammal'
Holonymy-Meronymy	Wholes to parts	*meecai* 'table' to *kaal* 'leg'
,,	Groups to members	*tuRai* 'department' to *peeraaciriyar* 'professor'
Meronymy-Holonymy	Parts to wholes	*cakkaram* 'wheel' to *vaNTi* 'cart'
,,	Members to groups	*paTaittlaivar* 'captain' to *paTai* 'army'

Opposites	Antonymic (gradable)	*ndallavan* 'good person' to *keTTavan* 'bad person'
,,	Complementary	*iravu* 'night' *to pakal* 'day'
,,	Privative (opposing features)	*ahRiNai* 'irrational' to *uyartiNai* 'rational'
,,	Equipollent (positive features)	*aaN* 'male' to *peN* 'female'
,,	Reciprocal Social roles	*maruttuvar* 'doctor' to *ndooyaaLi* 'patient'
,,	Kinship Relations	*ammaa* 'mother' to *makaL* 'daughter'
,,	Temporal Relations	*munnar* 'before' to *pinnar* 'after'
,,	Orthogonal or perpendicular	*vaTakku* 'north' to *kizakku* 'east' and *meeRku* 'west'
,,	Antipodal Opposition	*vaTakku* 'north' to *teRku* 'south'
Multiple opposites	Serial	*onRu* 'one', *iraNTu* 'two', *muunRu* 'three', *ndaanku* 'four'
,,	Cycle	*njaayiRu* 'Sunday' to *tingkaL* 'Monday' .. to *cani* 'Saturday'
Lexical association	Collocation	*cingkam* 'lion' to *karji* 'roar'
,,	Morphological relations	*paTi 'study'* to *paTittavan* 'educated man'
Compatibility		*ndaay* 'dog' to *cellappiraaNi* 'pet'

Table of lexical/semantic relations for verbs:

Relations	**Definition/sub types**	**Example**
Synonymy	Replaceable events	*tuungku* 'sleep' → *uRangku* 'sleep'
Meronymy-Holonymy	Events to super-ordinate events	*paRa* 'fly' → *pirayaaNi* 'travel'
Troponymy	Events to their subtypes	*ndaTa* 'walk' → *ndoNTu* 'limp'
Entailment	Events to the events they entail	*kuRaTTaiviTu* 'snore' → *tuungku* 'sleep'
,,	Event to its cause	*uyar* 'rise' → *uyarttu* 'raise'
,,	Event to its presupposed event	*vel* 'succeed' → *muyal* 'try'
,,	Event to its implied event	*kol* 'murder' → *iRa* 'die'
Antonym	Opposites	*kuuTu* 'increase' → *kuRai* 'decrease'

,,	Conversensess	*vil* 'sell' → *vaangku* 'buy'
,,	Directional opposites	*puRappaTu* 'start' → *vandtuceer* 'reach'

Chitiraputhiran's Tamil Lexical Resource: An electronic Data Base: Chitiraputhiran has taken up a major project supported by UGC on Preparing Lexical Resource for Tamil. He tries to capture polysemy and inter and intra relations between lexical items by preparing a database. He is building his database based on dictionaries and lexicons available for Tamil. So far, he has colleted lexical information for 2000 lexical items.

3.3. Electronic/Online thesaurus

There are attempts to prepare electronic thesauri for Tamil. One such attempt is mentioned below:

Tamil University's Electronic Thesaurus for Tamil: Rajendan and Baskaran of Tamil University have prepared an electronic thesaurus for Tamil.

The electronic thesaurus is based on a paper thesaurus prepared by Rajendran for Tamil (Rajendran, 2001. *taRkaalat tamizhc coRkaLanjciayam*, Tamil University: Thanjavur). The classification is based on Nida's theory of componential analysis of meaning (Nida 1975a & 1975b) and the lexical and semantic relations between the lexical items are established based on John Lyon's structural semantics (Lyons, 1977). The theory of field semantics (Lehrar, A. 1974. Semantic Fields and Lexical Structure, Amsterdam: North-Holland Publishing Company) also taken into consideration while preparing the thesaurus.

The preparation of electronic thesaurus has some linguistics issues like hierarchical classification of lexical items, establishment of semantics domains, selection of lexical items, establishing network of relations between lexical items, classification and ordering of lexical items under terminal domains etc. Computerization of thesaurus needs some procedures and methods for creating a special kind of databases and for accessing system.

The thesaurus is prepared in roman script and provided facility to display the content in Tamil script also. It contains nearly thirty thousand words. They have been classified into four major domains: entities, events, abstracts and relationals. The entities consists of mainly words denoting concrete nouns. The events consists of verbs and verbal nouns, the abstract consists of abstract nouns, adjectives and adverbs, and the relationals consists of coordinators, complementizers, postpositions, case suffixes and anaphoric references like pronouns. The lexical items are arranged hierarchically to capture hypernymy-hyponymy and holonymy-meronymy relations. Also the lexical items are arranged in the horizontal axis to capture horizontal relations such as synonymy,

lexical oppositions and lexical associations. Thus the lexical items are arranged in such a way that one can capture the network of lexical/semantic relations between lexical items.

3.4. Morphological Analyzers/Generators

Tamil is a Dravidian language. It is a verb final, relatively free-word order and morphologically rich language. Like other Dravidian languages, Tamil is agglutinative. Computationally, each root word can take a few thousand inflected word-forms, out of which only a few hundred will exist in a typical corpus. Subject-verb argument is required for the grammaticality of a Tamil sentence. Tamil allows subject and object drop as well as verb less sentences. In addition, the subject of a sentence or a clause can be a possessive Noun Phrase (NP) or an NP in nominative or dative case. As Tamil is an agglutinative language, each root word can combine with multiple morphemes to generate word forms. For the purpose of analysis of such inflectionally rich languages, the root and the morphemes of each word has to be identified.

The global scenario has influenced the morphological analysis of Tamil. In the last decade, computational morphology has advanced further towards real-life applications than most other sub-fields of natural language processing. To build a syntactic representation of the input sentence, a parser must map each word in the text to some canonical representation and recognize its morphological properties. The combination of a surface form and its analysis as a canonical form and inflection is called a lemma. The main problems are:

1. Morphological alternations: the same morpheme may be realized in different ways depending on the context.

2. Morphotactics: stems, affixes, and parts of compounds do not combine freely, a morphological analyzer needs to know what arrangements are valid.

A popular approach to 1 is the **cut-and-paste method**. The canonical form is derived by removing from and adding letters to the end of a string. The use of finite-state technology for automatic recognition and generation of word forms was introduced in the early 1980s. It is based on the observation that rules for morphological alternations can be implemented by finite-state transducers. It was also widely recognized that possible combinations of stems and affixes can be encoded as a finite-state network. An automaton containing inflected word forms can be upgraded to a morphological analyzer, for example, by adding a code to the end of the inflected form that triggers some predefined cut-and-paste operation to produce the lemma. Instead of cutting and pasting it at runtime, the entire lemma can be computed in advance and stored as a finite-state transducer whose arcs are labeled by a pair of forms. The transducer format has the advantage that it can be used for generation as well as analysis. The number of

nodes in this type of network is small, but the number of arc-label pairs is very large as there is one symbol for each morpheme-allomorph pair. A more optimal lexical transducer can be developed by constructing a finite-state network of lexical forms, augmented with inflectional tags, and composing it with a set of rule transducers. Lexical transducers can be constructed from descriptions containing any number of levels. This facilitates the description of phenomena that are difficult to describe within the constraints of the two-level model. Because lexical transducers are bidirectional, they are generally non-deterministic in both directions. If a system is only to be used for analysis, a simple finite-state network derived just for that purpose may be faster to operate.

A number of morphological analyzers and generators are prepared for Tamil. Morphological analyzers and generators for Tamil could be used for automatic lemmatization of word forms from corpus. The following is the list of computational morphological analysis attempted and/or implemented for Tamil:

Rajendran's Morphological Analyzer for Tamil: The first step towards a preparation of morphological analyzer for Tamil was initiated by *anusaraka* group of researchers under whose guidance Rajendran from Tamil University prepared a morphological analyzer for Tamil for Translating Tamil into Hindi at the word level.

Geanesan's Morphological Analyzer for Tamil: Ganesan developed a morphological analyzer for Tamil to analyze CIIL corpus. Now he is involved in improving his morphological parser.

Kapilan's Morphological Analyzer for Tamil Verbal Forms: Kapilan prepared a morphological analyzer for verbal forms in Tamil.

Vasu Ranganathan's Tagtamil: Tagtamil by Vasu Ranganathan is based on Lexical phonological approach. Tagtamil does morphotactics of morphological processing of verbs by using index method. Tagtamil does both tagging and generation.

AUKBCRC's Morphological Parser for Tamil: AUKBC NLP team under the supervision of Rajendran prepared a Morphological parser for Tamil. The API Processor of AUKBC makes use of the finite state machinery like PCKimmo. It parses, but does not generate.

Vaishnavi's Morphological Generator for Tamil: Vaishnavi researched for her M.Phil. dissertation on morphological generator for Tamil. Vaishanvi's morphological generator implements the item and process model of linguistic description. The generator works by the synthesis method of PCKimmo.

Winston Cruz's Parsing and Generation of Tamil Verbs: Winston Cruz makes use of GSmorph method for parsing Tamil verbs. GSmorph too does morphotactics by indexing. The algorithm simply looks up two files to see if the indices match or not. The processor generates as many forms as it parses and uses only two files.

Vaishnavi's Morphological Analyzer for Tamil: Vaishnavi again researched for her Ph.D. dissertation on the preparation of Morphological Analyzer for Tamil. She proposes a hybrid model for Tamil. It finds its theoretical basis in a blend of IA and IP models of morphology. It constitutes an in-built lexicon and involves a decomposition of words in terms of morphemes within the model to realize surface well-formed words-forms. The functioning can be described as defining a transformation depending on the morphemic nature of the word stem. The analysis involves a scanning of the string from the right to left periphery scanning each suffix at a time stripping it, and reconstructing the rest of the word with the aid of phonological and mophophonemic ruels exemplified in each instance. This goes on till the string is exhausted. For the sake of comparison she implements AMPLE and KIMMO models. She also evaluates TAGTAMIL, API Analyzer, and GSMorph. She concludes that Hybrid model is more efficient that the rest of the models.

Dhurai Pandi's Morphological Generator and Parsing Engine for Tamil Verb Forms: It is a full-fledged morphological generator and a parsing engine on verb patterns in modern Tamil.

4. SPELL CHECKERS/ GRAMMAR CHECKERS / STYLE CHECKERS

There are attempts to make spell and grammar checkers for Tamil. The following are the important ones:

Tamil University's Spell and Grammar Checker Project: Department of Linguistics of Tamil University under took a major project entitled 'Spell and Grammar Checker for Tamil' with the financial support of UGC. A morphological analyzer has been prepared to help in spell checking. An error analysis of written documents in Tamil has been undertaken and the errors are listed and classified. A model of Spell and grammar checker for Tamil with limited scope has been prepared.

Deivasundaram's Spell and Grammar Checker for Tamil: Deivasundaram is developing spell and grammar checker for Tamil to be incorporated in his Tamil Word Processor.

5. PARSING SYSTEMS

Parser is a device that makes use of the representation of the knowledge of the structure of a language provided by the grammar. It analyzes the input language structures and provides a parsed structure as out put. This could be

used for understanding and comprehending the structure and for translating the structures if necessary. The efficiency of the parser depends upon the effective database and delicate algorithms provided to the machine. In principle what a grammar declares, a parser applies for parsing and analysis.

5.1. Phonological

I have not found any serious work on Tamil in this area.

5.2. Morphological

Morphological processing as such forms the basis of any NLP system. Parsing is that activity where the analyzer recognizes and analyzes the given output and normally returns its along with its meaning and grammatical features as output. Generation is reversal of this process in which the root is combined with various morphemes to produce one or more surface forms. Most processors available have mostly based themselves to a structural approach. A number of models are available for morphological parsing. Important of them are PCKimo, Ample and GS Morph. PCKimmo uses two level morphology. Ample and GS Morph are based on item and arrangement. PCKimmo is tested and proved to have worked well for agglutinative languages like Finnish and Tukish at least. Ample is a morphological analyzer based on item and arrangement. It works by 'matching and filtering'. For the parsing, Ample starts form the left periphery of a word given for the analysis. The morphological parsing systems in Tamil culminate into number morphological processors that have been discussed under the heading 'Morphological analyzers/generators'.

5.3. Syntactic

A number of researches aim at preparing a syntactic parser for Tamil. Especially those who are trying for machine Translation across Tamil invariably build a syntactic parsing system at the shallow level or deeper level. The following are worth mentioning here:

Baskaran's Finite-state Machine for Syntactic Parsing: Finite-state Automata is one of the important techniques for parsing at all the level of a language structure. On experimental basis Baskaran (1984) has attempted a Finite-State-Machine for parsing sentences in Tamil.

Kumara Shanmugam's Syntactic Parser for Tamil: Phrase Structure grammars have been designed on fixed word order languages like English. Tamil is a variable word order language. In a sentence features of words or grammatical constituents can be tightly coupled or loosely coupled.

In fixed word order languages features like number and gender in the case of nouns and tense and number in the case of verbs are tightly coupled

attachments to the respective syntactic category. The other linkages are loosely coupled and indicated by word proximity.

In Tamil in addition to features like number, gender and tense, case attachments of nouns and aspect and mood of verbs are tightly coupled through inflectional attachments and do not need word proximity to indicate dependency. So Tamil requires a different kind of grammatical formalism which rely on dependency rather than proximity. Tamil rely more on morphology than syntax in indicating grammatical functions.

Keeping these characteristics of Tamil in mind Kumara Shanmugam (2004) has prepared a parser for Tamil. The parser he has designed carry out a complete morphological analysis of words of the sentences at the first level in order to help in dependency determination. The parser divides the sentences into two basic constituents noun part and verb part. In other words he has a one-level syntax tree. Since Tamil has variable word order it is possible that the noun or verb parts could be discontinuous. Thus the parser uses the morphological analyzer to determine tightly coupled features, which help in classification of the words. Unclassified words are classified based on heuristics. Dependencies between noun head and verb head and their respective modifiers are tackled with the help of dependency rules. Sentence patterns are then used to analyze the sentence. The selection of the sentence pattern depends on information provided by the morphological analyzer. The addition of rules for semantic dependencies can enhance the performance of the parser.

Shanmugam's Parsing Techniques: For processing a natural language certain formalisms are required. The grammatical models proposed by linguists, otherwise called as grammatical formalism try to capture the phonological, grammatical and semantic organization of natural language partially or fully. Grammatical formalisms are written with the purpose of comprehending the units and patterns found in all the levels of language. Computer scientists take the grammatical formalisms, modify them suitably for creating data-base procedures for machines so to make the machines process, recognize and produce natural language units and structures. Such a computational description is called as computational formalism.

Shanmugam while proposing a program for syntactic parsing in Tamil makes the following comments: "Structural description of the units of a languge can be provided by the grammar of language, making use of a principle called 'Projecton Principle'. According to this principle, as in transformational grammatical treatise, the structure of a sentence or phrase can be projected or plotted from the lexical specification of the head of the phrase or sentence. That projected structure will be abstract structure which will be modified with due substitution of appropriate lexical items.

Shanmugam (2002) advocates for minimalist program for Tamil parsing. All grammatical formalisms identify lexicon and certain procedures for creating and manipulating grammatical structures. Minimalist program which is a grammatical model and an extension of GB framework was proposed by Chomsky to expose the grammatical patterns found in languages. Some of his MPhil and Ph.D. students have worked for their dissertation on Context Free Grammar Formalism, Transformational Generative Grammar Formalism, Projection Principle, and Minimalist Program and prepared syntactic parser models for Tamil based on the formalism they have chosen.

Shallow Parsing in Tamil

We use the term shallow syntax as a generic term for analyses that are less complete than the output from a conventional parser. The output from a shallow analysis is not a phrase-structure tree. A shallow analyzer may identify some phrasal constituents, such as noun phrases, without indicating their internal structure and their function in the sentence. Another type of shallow analysis identifies the functional role of some of the words, such as the main verb, and its direct arguments. Systems for shallow parsing normally work on top of morphological analysis and disambiguation. The basic purpose is to infer as much syntactic structure as possible from the lemmata, morphological information, and word order configuration at hand. Typically, shallow parsing aims at detecting phrases and basic head/modifier relations. A shared concern of many shallow parsers is the application to large text corpora. Frequently partial analyses are allowed if the parser is not potent enough to resolve all problems. Church (1988) has designed a stochastic program for locating simple noun phrases which are identified by inserting appropriate brackets, [...].

Abney (1991) is credited with being the first to argue for the relevance of shallow parsing, both from the point of view of psycholinguistic evidence and from the point of view of practical applications. His own approach used hand-crafted cascaded Finite State Transducers to get at a shallow parse. Typical modules within a shallow parser architecture include the following:

1. Part-of-Speech Tagging. Given a word and its context, decide what the correct morphosyntactic class of that word is (noun, verb, etc.). POS tagging is a well-understood problem in NLP, to which machine learning approaches are routinely applied.

2. Chunking. Given the words and their morphosyntactic class, decide which words can be grouped as chunks (noun phrases, verb phrases, complete clauses, etc.)

3. Relation Finding. Given the chunks in a sentence, decide which relations they have with the main verb (subject, object, location, etc.).

Because shallow parsers have to deal with natural languages in their entirety, they are large, and frequently contain thousands of rules (or rule analogues). These rule sets also tend to be largely `soft', in that exceptions abound. Building shallow parsers is therefore a labour-intensive task. Unsurprisingly, shallow parsers are usually automatically built, using techniques originating within the machine learning (or statistical) community.

Parts of Speech Tagging in Tamil

Parts of speech tagging scheme tags a word with its parts of speech in a sentence. It is done in three stages: pre-editing, automatic tag assignment, and manual post-editing. In pre-editing, corpus is converted to a suitable format to assign a part of speech tag to each word or word combination. Because of orthographic similarity one word may have several possible POS tags. After initial assignment of possible POS, words are manually corrected to disambiguate words in texts.

Vasu Ranganathan's Tagtamil: Tagtamil by Vasu Ranganathan is based on Lexical phonological approach. Tagtamil does morphotactics of morphological processing of verbs by using index method. Tagtamil does both tagging and generation.

Ganesan's POS tagger: Ganesan has prepared a POS tagger for Tamil. His tagger works well in CIIL Corpus. Its efficiency in other corpora has to be tested. He has a rich tagset for Tamil. He tagged a portion of CIIL corpus by using a dictionary as well as a morphological analyzer. He corrected it manually and trained the rest of the corpus with it. The taggs are added morpheme by morpheme.

vandtavan: va_IV_ ndt_PT_avan_3PMS
pukkaLai : puu_N_PL_AC

kathambam of RCILTS-Tamil: Kathambam attaches parts of speech tags to the words of a given Tamil document. It uses heuristic rules based on Tamil linguistics for tagging and does not use either the dictionary or the morphological analyzer. It gives 80% efficiency for large documents. It uses 12 heuristic rules. It identifies the tags based on PNG, tense and case markers. Standalone words are checked with the lists stored in the tagger. It uses 'Fill in rule' to tag 'unknown words. It uses bigram and identifies the unknown word using the previous word category.

AUKBCRC's Parts of Speech Tagger for Tamil: A hybrid POS tagger is proposed for Tamil. This is a combination of a HMM based statistical POS tagger and a Rule based POS tagger. The system basically works as follows. First, the HMM tagger is trained using a small annotated corpus. Then new sentences are given and they are tagged. There may be untagged sentence due to the

limitations of algorithm and the amount of corpus used for training. Those sentences or words, which are not tagged

Chunking in Tamil

Basically a chunker divides a sentence into its major-non-overlapping phrases and attaches a label to each. Chunker differ in terms of their precise output and the way in which a chunk is defined. Many do more than just simple chunking. Others just find NPs. Chunking falls between tagging (which is feasible but sometimes of limited use) and full parsing (which more useful but is difficult on unrestricted text and may result in massive ambiguity. The structure of individual chunks is fairly easy to describe, while relations between chunks are harder and more dependent on individual lexical properties. So chunking is a compromise between the currently available and the ideal processing output. Chunkers tokenise and tag the sentence. Most chunkers simply use the information in tags, but others look at actual words.

Noun Phrase Chunking in Tamil

Noun phrase chunking deals with extracting the noun phrases from a sentence. While NP chunking is much simpler than parsing, it is still a challenging task to build an accurate and very efficient NP chunker. The importance of NP chunking derives from the fact that it is used in many applications.

Noun phrases can be used as a pre-processing tool before parsing the text. Due to the high ambiguity of the natural language exact parsing of the text may become very complex. In these cases chunking can be used as a pre-processing tool to partially resolve these ambiguities. Noun phrases can be used in Information Retrieval systems. In this application the chunking can be used to retrieve the data's from the documents depending on the chunks rather than the words. In particular nouns and noun phrases are more useful for retrieval and extraction purposes. Most of the recent work on machine translation use texts in two languages (parallel corpora) to derive useful transfer patterns. Noun phrases also have applications in aligning of text in parallel corpora. The sentences in the parallel corpora can be aligned by using the chunk information and by relating the chunks in the source and the target language. This can be done lot more easily than doing word alignment between the texts of the two languages. Further noun phrases that are chunked can also be used in other applications where in depth parsing of the data is not necessary.

AUKBCRC's Noun Phrase Chunker for Tamil : The approach is a rule based one. In this method initially a corpus is taken and it is divided into two or more sets. One of these divided sets is used as the training data. The training data set is taken and manually chunked for noun phrases, thus evolving rules that can be applied to separate the noun phrases in a sentence. These rules serve as the

base for chunking. The chunker program uses these rules and chunks the test data. The coverage of these rules is tested with this test data set. Precision and recall are calculated for this and the result is analyzed to check, if more rules are needed to improve the coverage of the system. If more rules are needed then additional rules are added and the same process as mentioned above is repeated to check for increase in the precision and recall of the system. The system is then tested for various other applications.

Vaanavil of RCILTS-Tamil: vaanavil identifies the syntactic constituents of a Tamil sentence. It ouputs the parsed tree in a list form. It tackles both simple and complex sentences. Simple sentences can have a verb, many noun phrase, simple adverbs and adjectives. Complex sentences can have multiple adjectival, adverbial and noun clausal forms. In the case of sentences with multiple clauses, vaanavil syntactically groups the clauses based on the cue words and phrases. It makes of phrase structure grammar. It uses look-ahead to handle free word order. It handles ambiguity using 15 heuristic rules. It uses the morphological analyzer to obtain the root word.

6. MACHINE TRANSLATION AND TRANSLATION TOOLS

Attempts for machine translation is the primary interest in NLP. Many attempts have been made for preparing machine translation systems across languages. Majority of the attempts aims at transferring information from English to Tamil. The following works need to be mentioned here:

Tamil University Machine Translation system for Russian-Tamil: On experimental basis, Departments of Computer Science, Linguistics and Translation of Tamil University, Thanjavur have developed a Machine Translation System for translating technical literature form Russain to Tamil. The research team had achieved a considerable success in the venture and a monograph entitled “Tamil University Machine Translation System” has been published.

Tamil-Hindi *Anusaraka*: Dr. Rajeev Sangal who was working in Department of Computer Science and Engineering, IIT, Kanpur under the financial support of DOE took up a major project for preparing translation aids (*Anusaraka*) to translate Indian languages from one to other using computer. The translation is at the word level as it presumes that the word order of Indian languages is more or less same and that the grammatical function depends more on inflection than word order.

The commonness between Indian Languages at the syntactic level has been exploited. *Anusaraka* works without a syntactic parser. There is only an efficient morphological analyzer. Rajendran of Tamil University Thanjavur associated with the group to build Tamil-Hindi *Anusaraka*.

A morphological analyzer was prepared for Tamil and the equivalent Hindi forms are given. The analyzed word structure of Tamil was mapped against their Hindi equivalents. The transfer relied on a transfer dictionary of Tamil-Hindi. A rudimentary Tamil-Hindi *Anusaraka* was developed in 1994. The NLP AUKBC Research Centre got all these materials Tami-Hindi *Anusaraka* Rajeev Sangal and started improving on it. The team came out with an improved version of the above-mentioned device.

Machine Translation Aid to translate Linguistics texts in English into Tamil: The demand for teaching Linguistics in Tamil has made it mandatory to look for a tool which helps in translating English Text books in English into Tamil. So a project with an aim to prepare a Machine Translation Aid (MTA) to translate Linguistics texts in English into Tamil was visualized by Rajendran, Department of Linguistics, Tamil University, Thanjavur. This work had culminated into a dissertation by Kamakshi (2001) which was published in book form (Kamaskhi & Rajendran, 2004). The project has the following objectives:

- To understand the different machine translation models which are in vogue for selecting a feasible model.
- To study the language of linguistics so that the domain specific features of the language of linguistics is thoroughly understood before proceeding to translate the linguistics texts by using machine.
- To correlate the structure of the source language, English and target language, Tamil so that the transfer model adopted for translation can be successfully manipulated.
- To prepare a prototype of Machine Translation Aid (MTA) to transfer linguistics texts in English into Tamil

The preliminary steps adopted to prepare the MTA are:

1. Understanding the architecture of English structure
2. Understanding the architecture of English structure
3. Correlation of English Structure with that of Tamil to find out the salient commonness and differences
4. Listing of transfer rules
5. Studying of style of Linguistics with special reference to Chomsky's Aspects of theory of Syntax
6. Preparation of a bilingual transfer dictionary

A MTA model has been built to transfer Linguistics Text in English into Tamil.

UNL-Interlingual Machine Translation approach for Tamil: This project has been under taken by the RCILRST Chennai. The Universal Networking Language (UNL) has been used as the intermediate representation. The device has an EnConverter and DeConverter. EnConverter is a language dependent parser that provides synchronously a framework for morphological, syntactic and

semantic analysis. EnConverter generates UNL expressions from sentences (or list of words of sentences) of a Tamil language by applying enConversion rules. In addition to the fundamental function of enconversion, it checks the formats of rules, and ouputs the messages for any errors. It also oupputs the information required for each stage of conversion in different levels.

With these facilities, a rule developer can easily develop and improve rules by using Enconverter. DeConverter is also a language dependent generator that provides synchronously a framework for word selection, morphological and syntactic generation and natural collocation necessary to form a sentence. Deconverter can convert UNL expressions into a variety of native languages, using a language specific set of word dictionary, grammatical rules and co-occurrence dictionary.

Given a set of structures the primary task is to retrieve the relevant dictionary entries from the Tamil language word dictionary corresponding to the words in the word part of the UNL structures.

The next step in the DeConversion process is use of specific language specific, linguistic based deconversion rules to convert the UNL structure into natural language sentences. These sentences have to obey the morphological and syntactic rules of the language. This is ensured by appropriately building the deconversion ruels which specify the morphological syntactic structure of the language under consideration.

Vasu Renganathan's Interactive Approach to Development of English-Tamil Machine Translation System on Web: The work-in-progress of this system may be tested online in URL http://lrrc3.plc.upenn.edu/tamil/. This is rule based system containing around five thousand words in lexicon, and a wide range of transfer rules written in Prolog encompassing frequently occurring English structures mapped to corresponding Tamil structures. Both rule base and lexicon of this system are built in such a way that the users can update the scope of this system interactively by adding words into lexicon and rules into rule base. Translating both colloquial and technical English into Tamil with a computer essentially involves construction of the two blocks namely lexicon and rules into rule-base.

Durai Pandi's English-Tamil Machine Translation System: A working model of English to Tamil Machine Aided Translation package for one finite verb (simple sentences) structures has been designed based on TAM encoded MAT compatible lexicon, English and Tamil structure parsing engine and a Tamil structure generator engine. Presently the package is under development with the assistance form Tamil Software Development Fund (TSDF) of Tamil Virtual University (TVU), Chennai.

1. Translation memories

There are a few works in this area of research. Specific references are not available.

2. Terminology data books

Radha Chellappan has a developed a terminology data book. Specific reference is not available.

3. Post-editing

There are for post-editing packages for the translation output in Tamil. Specific references cannot be given.

4. Word Sense Disambiguation (WSD) tools

AUKBCRC's Word Sense Disambiguation Tool: Word sense disambiguation is the task of assigning the appropriate sense for all the occurrences of ambiguous words in the text. Most of the WSD systems use the context of the ambiguous word to determine its intended usage. Baskaran worked for his MS degree on word sense disambiguation in Tamil. Baskaran (Baskaran 2003, 2004) has prepared a WSD tool for Tamil to help in MT.

7. OPTICAL CHARACTER RECOGNITION (OCR)

7.1. Single Font/Multifont/Omnifont OCR systems

Traditionally, data are keyed into a computer through a data entry operator to store the information. While it is still used, and is the most reliable method, a need for automating this process was required, in more labor intensive tasks such as: mail sorting, passport processing systems, Insurance and Finance Forms processing systems, Bill Processing systems and many such voluminous sifting in a diminutive amount of time. To automate these processes, character and handwriting recognitions techniques are used.

Krishnamoorthy's OCR System: Krishnamoorthy is seems to be the torch bearer in the race of preparing OCR softwares for Tamil. There are many approaches in designing OCR. Krishnamoorthy has chosen a new method, based on representing a letter as a graph. If a letter is considered as strokes of thing lines, it can be considered as a graph, by inserting vertices. These vertices can be the end points, points which are local minimum or local maximum in the x and y directions. This representation of a letter as a graph has some advantage. The major advantage is that the information content gets reduced very much. Hence the processing needed to recognize a character gets reduced very much, in many cases. This speeds up the recognition processes very much.

Krishnamoorthy has devised a method in which this graph is constructed without thinning the character map. This again adds to the speeding up of the recognition.

The major disadvantage, according to Krishnamoorthy, is that some times too much information gets lost, and he has to resort to different methods to recognize a letter. He has listed a number problems and solutions to overcome these problems. According to him the solutions are complicated. They will increase the time recognition very much. But, if accuracy required is more than 95% or so, it may be necessary to employ all these techniques. Also, the logic should be built in such as way that each solution is handled intelligently so that the time taken is small. The shape of the characters, the linguistic nature of words, and the different approaches for character recognition - all these have to be mixed and used judiciously to get the best result in OCR.

7.2. Printed / typed / handwritten /shorthand

There are a number of research works in these areas of study for Tamil. A few are listed below:

Suresh et al's approach on Recognition of Hand written Tamil Characters: Suresh, Arumugan and Aravanan (2002) propose fuzzy classificatory approach for recognition of handwritten Tamil characters.

Hewvitharana and Fernando's Two stage Classification Approach to Tamil Handwriting Recognition: Hewvitharana and Fernando (Hewvitharana and Fernando, 2002) propose a system to recognize handwritten Tamil characters using a two-stage classification approach, for a subset of Tamil alphabet. In the first stage, an unknown character is pre-classified into one of the three groups: core, ascending and descending characters. Then, in the second stage, members of the pre-classified group are further analyzed using a statistical classifier for final recognition. Hewvitharana and Fernando claim that a recognition rate of 80% was achieved for the 1st choice and 97% for the top 3 choices.

Vasudevan's Character Recognition Techniques: Vasudevan (Vasudevan, 2002) proposes character recognition techniques for Tamil.

7.3. Online/ Off line

There are attempts to recognize characters at the online and off line too. Specific reference is missing.

8. INFORMATION RETRIEVAL /INFORMATION EXTRACTION

A number of research works have been undertaken for information retrieval and information extraction for Tamil. AUKBCRC and AU RCILTS-T have developed advance information retrieval and information extraction systems for Tamil. There are attempts for new aggregation too.

8.1. Text mining

AUKBCRC and Anna University RCILTS-T have prepared a tool for text mining. Specific reference is missing.

8.2. Web mining

AUKBCRC and Anna University RCILTS-T have prepared a tool for text mining. Specific reference is missing.

9. SEARCH ENGINES /WEB TECHNOLOGIES

There are search engines made for Tamil too.

10. SPEECH TECHNOLOGY

Many research works have been attempted and some are successfully completed on speech technology in Tamil. IIT Madras was/is involved in developing speech technology tools for Tamil.

10.1. Signal Processing

Natansapapathy's study of Segmental Duration Tami sounds: Natanasapabathy (2002) studies the segmental duration of Tamil sounds the knowledge of which is essential for speech synthesis. According to his analysis the timing process operates at least in three different levels:

- at the level of sentence, phrase, word, and syllable where it is related to the boundary phenomena represented by vowel lengthening.
- at the phonological level as contrastive distribution in both vowels and consonants.
- at the segmental phonetic level to accommodate positional and contextual effects

The relative duration is sufficient for phonological and linguistic purposes. However, for the purpose of speech synthesis, the absolute duration or at least three levels of duration is needed for the computer to understand the speech fully and correctly.

10.2. Text to speech (TTS)

Text-to-speech synthesis involves interdisciplinary work, covering language processing and speech processing. There are many works in this area of research. Some of them are noted here.

Yegnanarayanan's work on text-to-speech: Yegnanarayanan (Yegnanarayanan 1992, 1993, 1994) has made text-to-speech system for Indian Languages and also attempted speech synthesis machine.

***ethiroli*:** This is also a product of RCILTS-T, Chennai. *Ethiroli* is a Text-to-speech Engine for Tamil. The engine has been so designed that it can be plugged into any application requiring a text-to-speech output. It is Tamil language specific and can be used in systems like Content Packages, Chatterbots, Telephone Interfaces to Online Help Systems and Interactive Sale Computers. The important features of this device are:

- It handles ambiguity.
- Used Concatenation methods.
- Implements Cues and Silence techniques.
- Processes TAM encoded text files.

Prathibha et al's work on text-to-speech synthesis system: They have made two versions of their work, Thirukkural I and Thirukkural II. The second one is an improved version of the first one. They claim that Thirukkural II generates intelligible and acceptable natural speech. It also has the facility to produce different voices like female, child and old man. It can also read digits present in the text. The synthesized speech is rendered natural by incorporating prosodic rules. Currently they are attempting to synthesize emotions such as sadness, anger and joy.

10.3. Speech to text (STT)

There are a few attempts to convert speech into text for Tamil. This task requires extensive research in speech technology with regard to Tamil.

10.4. Speech Recognition / Understanding

Speech recognition aims at signal transformation. There are a number works which aims at this topic of research. Few are discussed below.

a) Language Recognition

Syllabic Study of Naymulla Kahan: For recognition in Indian languages syllable-like units seem to be an appropriate unit to achieve this transformation.

Nayemulla Khan and his team in their efforts to develop a language independent syllable recognizer, studied the basic characteristics of the words and syllable-like subword units occurring in three Indian languages (Tamil, Telugu and Hindi). The observation about the words and syllable like-units, as to their structure, duration and statistical properties are presented in the paper quoted below. The implication of these observations in the context of speech recognition and language identification is highlighted in the paper.

b) Speaker identification

A few researches have been attempted in this line of thinking for Tamil. There are attempts to identify speakers involved in telephonic conversation.

11. STANDARDIZATION ISSUES

A simple application like publishing can be implemented by just having fonts to compose Indian language text. If the text to be processed is of monolingual nature, mapping on to any existing coding can do it. However in bilingual, multilingual context a language text cannot be processed without identifying the language code. Therefore, encoding a language is very much necessary and is inevitable for the purpose outlined below:

- Makes easy to identify the language characters, there by simplifying the language processing complexities
- Easily intermix with any language.
- Eliminates the usage of mark up languages.

11.1. Character level standard: ISCII/UNICODE

In the computer world, R & D Institutions, Companies involved in technology development and its usage evolve a common specification called 'standard' to meet their varied requirements. Sometimes, the well-established protocols developed by the market leaders also evolve as a 'standard'. There are standards for every aspect necessitating its implicit implementation and its goes with natural language as well.

BIS is the nodal authority and it has standardized all Indian Languages for computer and notified as IS:13194:1991. Likewise, the Unicode Consortium defines Unicode as standard for all languages of world. The current version is Unicode 3.0. Characters of all languages can be interpreted by using Unicode. When all the OS softwares are using Unicode for information interchange, no language would turn into garbage (which is the common problem with today's implementation). Unicode is the only wayout to solve the language problems.

The present Unicode for Tamil, which is encoded as vowels, consonants and vowel signs, is based on earlier version of ISCII.

Shortcomings of Unicode for Tamil: Anparasan (2001) notes the following as the shortcomings of Unicode 3.0 for Tamil:

Encoding Tamil alphabet: Tamil alphabets are encoded as ayutham, vowels, consonants with *a* as and vowel sign.

- It is to be noted here that the matras are not part of character set of Tamil, which amount to redefining character sets of Tamil.
- Vowel Consonants are encoded instead of pure consonants.
- Anuswar is not a Tamil alphabet but encoded.

Order of Tamil alphabets: While encoding Tamil alphabets, Tamil linguistic alphabetic order, has not been followed.

Wrong interpretation: The following explanation is presented only to show the quantum of error and not to rectify the coding of mathras. The vowel matharas such as bh, Bh and bs are treated as if they are formed of two mathras and it is not correct. A vowel interpretation in a vowel consonant is identified by its allograph called mathras or signs, wherein the vowel consonant is formed of one consonant and one vowel. Similarly the case of xs cannot be formed of ‘o’ short and ‘au’ vowel sign. And also two ‘au’ signs are necessary, even in the present implementaion.

NLP: Natural Language Processing such as morphological analyzer, spell checker, grammar checker, translation etc entirely depend on the pure consonants. Encoding vowel consonant forces further processing apart from language identity which is unjustifiable.

Clash with Tamil Grammar: By encoding halant, the basic rule of forming vowel consonants is getting modified. As all of us know that the vowel consonants are formed from the basic consonants. As per the standard now the basic letter becomes the vowel consonant i.e. ‘a’ consonant, which leads to many linguistic analytical problems.

Standardization Efforts of Government of Tamilnadu

Government of Tamilnadu constituted a Committee to study the issues related to encoding schemes, keyboard layouts, technical words and to recommend suitable standards for developing Tamil on computers.

To deliberate, discuss and arrive at consensus on the technical issues, the first International Seminar was organized at NUS, Singapore, with the efforts made by (Late) Naa. Govindaswamy in 1997.

The second Intenational Seminar was organized by the Government of Tamilnadu in 1999 in Chennai. The following draft standards were announced at the valedictory function of the seminar:

- Standardized phonetic keyboard
- Bilingual Glyph encoding
- Monolingual Glyph encoding
- Character encoding

However, the first three draft standards were finalized and announced during June 1999. The character-encoding scheme was clubbed with Tamil Unicode standard.

Standardization Efforts of Government of India

Department of Electronics (DoE), Government of India, has announced the First Standard ISSII-83 for all Indian languages. Further, the ISSCII code was revised in 1988 to evolve IBM PC counterpart PC-ISCII. The existing ISCII standard was adopted by Bureau of Indian Standards and announced as IS:13194:1991. On understanding the requirement of rectifying existing standard DoE, has sent a Questionnaire to all State Government during June 2000 to clarify the points related to Unicode. In response to this communication, the Standardisation Committee constituted by Government of Tamilnadu recommended a 384 syllable encoding scheme for Tamil in Unicode which consists of various allographs and symbols apart from vowels, consonants, and vowel consonants.

There are three types of possibilities in encoding Tamil Unicode as under (Anparasan, 2001):

Vowels, Ayutham and pure consonants
Encode all letters, symbols, glyphs etc
Vowels, consonants and mathras

Anparasan (2001) points out that by encoding Tamil in its alphabets i.e. vowels, ayutham and consonants, the encoding would support the following:

- Represent a true Tamil alphabet system.
- can be used for any type of writing system
- Tamil can be implemented with just fonts
- Already OS, Database, Office suites are available with Tamil Unicode support.
- Uniscribe technology available to develop any specific application with support for Tamil.
- OTF supports complex Indic scripts including Tamil.
- Development of NLP applications can be further accelerated due to simplicity in handling language codes.
- Sorting and Indexing
- Less overhead when embedding fonts.
- Unique representation of language alphabets and letters.

11.2. Glyph standardization

There are a few works on this area of research. Muthu Nedumaran's (Muthu Nedumaran, 2002) work on 'Glyph choices and techniques for building Unicode' can be quoted as one of the works in this line of thinking.

11.3. Keyboard layout

It has been agued that though the classificatory systems of sounds of Tamil is based on their phonetic articulatory properties, the alphabetic system is not phonetic. The standard keyboard overly standardized on the basis of certain convenience lacks scientific objectivity. The assumption that Tamil consonant scripts have an implicit vowel *a* included in them, and arbitrary provision for deleting it in conjunct formation and in some other distributional contexts are not based on any phonetic consideration. It is argued that though the existing standard ISCII character codes for consonant script with supposedly implicit vowel *a* and the provision for deleting by a dot (halant) many serve visual representation of the consonant characters, it is technologically inadequate for NLP involving linguistic analysis. It is further argued that the uniform keyboard overlay for all Indian languages is arbitrary since the frequency count of an alphabet differs from language to language.

11.4. Rendering engines

Rendering engines have been attempted for Tamil.

11.5. Operating System level support

There are attempts to make operating system level support in Tamil. The following need to mentioned here:

PONN: A Tamil Operating Environment: In order to help Tamil speakers who do not know English or do not want to make use of English for computer operations Parasnna Venkatesan-and-team has proposed a Tamil operating environment. PONN has the following generic features:

- Open to different standards followed in different schemes in Tamil S/W development
- Platform independent
- Scalable
- Generic framework for Tamil Software development
- PONN Abstract Classes – reusable class library
- Set utilities like Shell, PONN Explorer, VASU etc.

PONN desktop provides the user interface by launching different utilities and dynamically choosing the working language of PONN as either Tamil or English. Shell is non-GUI for the PONN storage manager. This helps the user to carry out their file and directory operations in Tamil using keyboard. In contrast to the above, PONN Explorer is a GUI for the PONN storage manager that aids the user to visualize their files and directories, and operations on them are carried out using mouse clicks.

KURAL is a Tamil programming language. It is similar to any imperative programming language. It is designed for teaching programming in Tamil. A compiler is also designed for translating KURAL program into KURAL intermediate code. An execution unit also designed for executing these. KURAL Integrated Development Environment is created, which acts as an editor to create KURAL program and carry out the translation and execution of it.

VASU is a simple document editor designed to create documents in Tamil. Also help facility is provided with the environment.

PONN-A Tamil Operating Environment has been implemented using VC++ and Java and tested in Windows and Linux platforms.

Thenkoodu: RCILTST has developed a package called 'Thenkoodu' which supports MS-Assess format. Thenkoodu is a Tamil database package to help business groups to handle their data efficiently. It helps to store Tamil data and also provides various means of manipulating, processing and retrieving the data. The relevant data can be extracted form the database using queries, forms and reports. All the data base are stored in MS-Access format.

***Arangam*:** RCILTST is a presentation tool for Tamil. It helps to organize a presentation with slides consisting of text, pictures and images. Arangam is a necessity of computerized presentation in the technologically savvy world. The existing presentation software does not easily support Tamil language. It creates an affordable and a thin presentation package. Contemporary software has the

following deficiencies: written only for one operating system and highly priced for a single user. It has the following features:

- Supports add, remove or edit operations on objects.
- Has three modes viz. Edit, Preview and Presentation Mode.
- Edit mode allows correction on one selected slide.
- Preview mode gives a preview of many slides.
- Presentation mode is for slideshow.
- Supports slide background and font format.

***Chathurangam*:** This is also prepared by RCILTST. *Chathurangam* is a spreadsheet application that helps in financial applications and calculations. It has Tamil user interface and accepts information in Tamil. *Chathurangam* helps to edit or save the data and view the data using various charts easily Mathematical expressions are also handled. The salient features of this device are:

- Supports many data formats.
- Handles basic editing operations and mathematical expressions.
- Print facility is available.
- Different chars help to see the patterns in data.

11.6. Browser level support

Browser level support is made for Tamil. The following one package need to be mentioned here:

***Bavani*:** This is also a product of Anna University RCILTS-T, Channai. *Bavni* is a Tamil search engine for documents in the Internet. It searches for Tamil words in Tamil web sites available in popular font encoding schemes. An important feature of the search engine is the integration of the Morphological Analyzer. All searches are based on root words generated from the query given by the user. This allows a large coverage of documents in the Internet. It has the following salient features:

- Currently supports 22 font-encoding schemes.
- Allows search on multiple keywords.
- Supports search on English words.
- Has an elegant user interface to enter the query.

Conclusion

I have tried to list down the works undertaken by the scholars and institutions for the technological development of Tamil. I don't claim that the list is

exhaustive. There may be many lapses and omissions. I request the scholars who are involved in the technological development of Tamil to furnish me with the contributions not mentioned by me so that I can include them in my next version of the paper.

References

Anbarasan, N. 2001. "Tamil Unicode-What do we need?", paper read in National Seminar on Computational Linguistics and Dravidian Languages. CAS in Linguistics, Annamalai University, Annamalai.

Anandan, P, Rajani Parthasarathy, Geetha, T.V. 2001. "Morphological Generator for Tamil", in Proceedings of the Tamil Internet 2001 Conference.

Annamalai, E. "Corpora Development in Indian Languages", in Agarawal and Pani (eds.) Information Technology Applications in Language, Script and Speech, New Delhi: BPB Publication.

Arulmozhi, P., Sobha, L., Kumara Shanmugam, B. 2004. Parts of Speech Tagger for Tamil, Symposium on Indian Morphology, Phonology & Language Egnineering, Indian Institute of Technology, Kharagpur.

Arulmozhi, P and Sobha, L. 2006. A Hybrid POS tagger for a Relatively Free Word Order Language. In Proceedings of the First National Symposium on Modeling and Shallow Parsing of Indian Languages, pages 79-85.

Arulmozi, S. 1998. Aspects of Inflectional Morphology–A Computational Approach. Ph.D dissertation submitted to University of Hyderabad.

Arulmozi, S and M.C. Kesava Murty. 2006. Verb Sense Disambiguation in Telugu-Tamil MT. In Proceedings of the First National Symposium on Modeling and Shallow Parsing of Indian Languages pages 107-111.

Balakrishnan, R. 2002. Morphology and Tamil Computing. Paper read in International Seminar on Tamil Computing, February 27, 28, 2002, Madras University.

Baskaran, S. "User Interface with Computers in Tamil-an Overview [*kaNippoRi uraiyaakam*]", in Proceedings of the Third Tamil Nadu Science Congress, Pondicherry.

Baskaran, S. "Zip's Law: A perspective study in Tamil", in Proceedings of Eighth International Conference of Tamil Studies.

Baskaran, S. Report on Standardization of Tamil Key Board, Submitted to the Committee formed for Standardization of Tamil Key board by Tamilnadu Government.

Baskaran, S. "Tamil Computing", *tamizh vaLarcci*.

Baskaran, S. "Evolution of Tamil computing", Souvenir, National Symposium on Current Trends in Computer Applications.

Baskaran, S. "Experiment in Implementation of Finite-State-Machine Parser for Tamil Sentences", in Computer Society of India Communications, Bombay.

Baskaran, S. "Compute as a Language Research Tools," in Proceedings of the Second Tamilnadu Congress, Madras.

Baskaran, S. 2002. Word Sense Disambiguation of Tamil. MS dissertation submitted to Anna University, Chennai.

Baskaran, S. 2003. "Word Sense Disambiguation in Tamil", International Journal of Dravidian Linguistics.

Baskaran, S and Vaidehi, V. 2004. "Collocation Based Word Sense Disambiguation using Clustering for Tamil", International Journal of Dravidian Linguistics 33.1: 13-28.

Baskaran, S. Vijay-Shanker, K. 2003. "Influence of Morphology in Word Sense Disambiguation for Tamil", in Rajeev Sangal et al (eds.) Recent Advances in Natural Language Processing. Mysore: Central Institute of Indian Languages.

Brochure on 'Language Technology Products' of the Resource Centre for Indian Language Technology Solutions-Tamil, Chennai.

cevveeL kapilan. 1994. *kaNippoRivazhi tamizh vinaikaLin pakuuppaayvu. cennai: puttaakka mozhiyiyal kazhakam.*

Chellamuthu, K.C. et al. Tamil University Machine Translation System (TUMTS), Thanjavur: Tamil University.

Chellamuthu, K.C. 2002. 'Russian to Tamil Machine Translation System at Tamil University,' in Kalyansundaram K (ed.) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, 74-83.

Chithiraputhiran, H. Tamil Lexical Resource: An electronic Data Base. Paper read in International Seminar on Tamil Computing, February 27, 28. Madras University.

cuppaiyaa piLlai, 2000. *iyaRkai mozhi aayvu. cennai: ulattamizhaaraaycci ndiRuvanam.*

Deivasundaram, N. and Gopal, A. 2003. 'Computational Morphology of Tamil' In B. Ramakrishna Reddy (ed.) Word Structure in Dravidian, Kuppam: Dravidian University, 406-410.

Dhurai Pandi, 2002. "Morphological Generator and Parsing Engine for Tamil Verb Forms (abstract)", in Kalyanasundaram (ed) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, page 59.

Durai Pandi. 2002. "English-Tamil Machine Translation System", in Kalyansundaram K (ed.) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, page 86.

Durai Pandi, 2002. "Morphophonemic Rules for Tamil Computing", in Kalyansundaram K (ed.) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, page 98.

Elangovan, A. "Optimisation Techniques in Unicode Tamil Font Development", in Tamil Internet 2002: Conference Papers, Cennai: Asian Printers, page 30.

Final Report: Development of Corpora of Texts of Indian Languages in Machine Readable form: Part II (Tamil, Telugu, Kannada, Malayalam), TDIL-Corpora Group, CIIL, Mysore January 1995.

Francis Ekka and Ganesan, M. 1994. "Issues on Standard ISCII Codes and Inscript Keyboard", Agarawal and Pani (eds.) Information Technology Applications in Language, Script and Speech, New Delhi: BPB Publication.

'The Unicode Standard: A Technical Introduction' 8/05/2001.

Ganesan, M. 1994. "Functions of the Morphological Analyser Developed at CIIL, Mysore", in Harikumar Basi (ed.) Automatic Automatic Translation (seminar proceedings), Thiruvananthapuram: ISDL.

Ganesan, M. 1994. "'A Scheme for Grammatical Tagging of Corpora of Indian Languages", in B.B. Rajpurahit (ed.) Technology and Languages, Mysore:CIIL

Ganesan, M. 1996. "Relevance on Indian Grammatical Theories for Automatic Translation: chances and challenges", paper presented in National Seminar of Dravidian Linguists Association, Dravidian University, Kuppam, June 1996.

Ganesan, M. 2000. "Compilation of Electronic Dictionary for Tamil", Annamalai University, Annamalai Nagar. www.bhasshaindia.cjb.net/G7ganesan.pdf

Ganesan, M. 2003. “Computational Morphology of Tamil”, in B. Ramakrishna Reddy (ed.) Word Structure in Dravidian, Kuppam: Dravidian University, pages 399-405.

Ganesan, M and Francis Ekka. 1994. “Morphological analyzer for Indian Languages”, Agarawal and Pani (eds.) Information Technology Applications in Language, Script and Speech, New Delhi: BPB Publication.

Hewavitharana, S. 2002. ‘A Two Stage Classification Approach to Tamil Handwriting Recognition’ in Kalyansundaram K (ed.) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, page 118-124.

Information Technology Department. ‘Information Technology – Standardization of Tamil Key Board and Encoding of Tamil. Glyphs – Recommendations of the Sub-committee on Tamil in Information Technology – Accepted – orders – Issued. (G.O. Ms.NO.17 dated” 13 June 1999)

Jayaram, B.D. “Development of Corpora in Indian Languages”, paper presented in the Seminar in the use of Computers in Indian Languages, held in CIIL, Mysore, August 1992.

Kalyanasundaram, K.A. “Comparison of transliteration schemes and key mapping of Tamil fonts”, http://www.Geocities.com/Athens/5180/translit.html 19/12/2000

Kamakshi, S. 2002. “Preliminaries for Digitizing the Personal Pronouns in English into Tamil (Distribution Sensitive Machine Translation Aid–DSMTA)–A Demo Paper”, in Kalyansundaram K (ed.) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, pages 87-97.

Kamakshi, S. 2004. “Machine Recognition and Translation of ‘~ing’ words in English into Tamil through bilingual Machine Tractable Dictionary”, paper presented in International Conference on Indian Lexicography 28-30 January 2004, CASL, Annamalai University, Annamalainagar.

Kamakshi, S. and Rajendran, S. 2004. Preliminaries to the preparation of a machine aid to translate Linguistics Texts written in English into Tamil. DLA Publications, Thiruvanathapuram.

Kasirao, V. “Impact of Information Technology on Information Management Sevices with Special Reference to Tamil Computing: A Study”, paper read in International Seminar on Tamil Computing, February 27, 28. Madras University.

Krishnamoorthy, V. “OCR Software for Printed Tamil Text”, in Kalyansundaram K (ed.) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, pages 99-101.

Kumara Shanmugam, B. 2001. “Parse Representation of Tamil Syntax”.

Kumara Shanmugam, B. 2002. “Machine Translation as related to Tamil”, in Kalyansundaram K (ed.) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, pages 84-85.

Kumara Shanmugam, B. 2004. Syntactic Parser for Tamil. M.S. (in NLP) dissertation submitted to Anna University, Chennai.

Lyons, J.1977. Semantics, volume 1, New York: Cambridge University Press.

Michael S. Kapalan. 2002. “Unicode and Tamil”, in Tamil Internet 2002: Conference Papers, Cennai: Asian Printers, page 1.

Muthu Nedumaran. 2002.”Glyph Choices and Techniques for Building Unicode Based Tamil Fonts”, in Tamil Internet 2002: Conference Papers, Cennai: Asian Printers, page 31.

Natanasabapathy, S. 2002. “Study of Segmental Duration Tami sounds”, paper presented in International Seminar on Tamil Computing, 27-28 February and 1 March, 2002, University of Madras, Chennai.

Nida, E.A. 1975.a. Compositional Analysis of Meaning: An Introduction to Semantic Structure. The Hague: Mouton.

Nida, E.A. 1975.b. Exploring Semantic Structure. The Hague: Mouton.

Navaneethan, P et al. 2004. “Multilingual Word Processor with Online Noun Translator”, paper read in National Seminar on Cross Language Communication Tools (Machine Translation) for Indian Languages, February 25-27, 2004.

Nayeemulla Khan, A, Suryakanth, V, Gangashetty, & Yegnanarayana, B. 2003. “Syllabic Properties of Thee Indian Languages: Implications for Speech Recognition and Language Identification”, in Rajeev Sangal, S.M. Bendre & Udaya Narayana Sigh (eds.), Recent Advances in Natural Language Processing. Mysore: CIIL

paaskaran, *ca*. 2004. *tamizhil kaNippoRiyiyal, kaNippoRiyiyiyal tamizh. tanjcaavuu: umaa patippakam.*

Ponnavaiko, M. 2002. “An Investigation on Unicode Standards for Tamil”, in Tamil Internet 2002: Conference Papers, Cennai: Asian Printers, page 29.

Prasanna Venkatesan, P and Chitrakekha T and Kuppuswami P. 2002. “PONN: A Tamil Operating Environment”, in Proceedings of Tamil Internet 2002 Conference, Cennai: Asian Printers.

Prathibha, P., Ramakrishana, A.G. 2002. “Web-enabled Speech Synthesizer for Tamil”, in Proceedings of Tamil Internet 2002 Conference, Cennai: Asian Printers, pages 134-140.

Prathibha, P., Ramakrishana, A.G. and Muralishankar, R. 2002. “Thirukural-II – A Text-to-speech Synthesis System”, in Proceedings of Tamil Internet 2002 Conference, Cennai: Asian Printers, 126-133.

Rajan, K and Ganesan, M. 2002. “Tools for Corpus Analysis”, paper presented in International Conference on South Asian Linguistics (ICOSAL-4), 3-5 December, 2002, CASL, Annamalai University, Annamalainagar.

Rajendran, S. 2002. “Preliminaries to the Preparation of wordnet for Tamil”, Language in India2:1, www.languageinindia.com

Rajendran, S. 2003. “Dravidian WordNet: a proposal”, in R.M. Sundaram, et al (eds) Facets of Language, Thanjavur

Rajendran, S. 2003. “Creating Generative Lexicon form Dictionaries: Tamil Experience”, in Rajeev Sangal, S.M. Bendre & Udaya Narayana Sigh (eds.), Recent Advances in Natural Language Processing. Mysore: CIIL

Rajendran, S. 2003. “Prerequisite for the Preparation of an Electronic Thesaurus for a Text Processor in Indian Languages”, Language in India 3:1, www.languageinindia.com

Rajendran S, Arulmozi S, Ramesh Kumar S, & Viswanathan S. 2003. “Computational Morpohology of Verbal Complex”, in B. Ramakrishna Reddy (ed.) Word Structure in Dravidian, Kuppam: Dravidian University, 376-398.

Rajendran, s. and Baskaran, S. 2002. “Preparation of Electronic Thesaurus for Tamil”, in Proceedings of the International Conference on Natural Language Processing. Mumbai: NCST.

Rajendran, S. and Kamakshi, S. 2002. “Preliminaries to the preparation of a machine translation aid to translate Linguistics Texts in English into Tamil”, Paper read in UGC-SAP National Seminar ‘On Translation’ 7th-9th March 2002, CAS in Linguistics, Annamalai University.

Ramakrishnan, A.G. “Issues in Standardization for Text to Speech in Tamil”, Tamilnet 2001, Kulalumpur, Malaysia.

Ramaswamy, V. 2000. Morphological Generator for Tamil. Unpublished M.Phil. dissertation. University of Hyderabad.

Ranganathan, V. 1997. “A Lexical Phonology Approach to Tamil Words by Computer”, International Journal of Dravidian Linguistics 26:1.57-70.

Shanmugan, C. 2001. “Computer Analysis of Simple Sentences in Tamil”, Paper read in UGC-SAP National Seminar on Computational Linguistics and Dravidian Languages, 22-24 February, 2001, CAS in Linguistics, Annamalai University, Annamalainagar.

Shanmugan, C. 2002. ”Grammar and Parser: A Program for Syntactic Parsing in Tamil”, International Seminar on Tamil Computing, 27-28 February and March 1, 2002, University of Madras, Chennai.

Shanmugan, C. “Minimalist Program for Tamil Parsing”.

Sobha, L and Vijay Sundar Ram. 2006. “Noun Phrase Chunker for Tamil Language”, in Proceedings of the First National Symposium on Modeling and Shallow Paring of Indian Languages, pages 194-198.

Suresh, R.M., Arumugan, S., and Aravanan, K.P. 2000. “Recognition of Handwritten Tamil characters using fuzzy classificatory approach”, in Proceedings of the Tamil Internet 2000 Conference, Singapore.

Vaishnavi Ramaswamy. 2000. A Morphological Generator for Tamil. M.Phil Dissertaion Subbmitted to University of Hyderabad.

Vaishnavi Ramaswamy. 2003. A Morphological Analyzer for Tamil. Ph.D Dissertaion Subbmitted to University of Hyderabad.

Vaishnavi, Ramaswamy. 2003. “Parsing in AMPLE, KIMMO & PERL: Nouns in Tamil”

Vasudevan V. 2002. “Character Recognition Techniques–A Demo Program”, International Seminar on Tamil Computing, 27-28 February and March 1, 2002, University of Madras, Chennai.

Vasu Renganathan. 2002. “Interactive Approach to Development of English-Tamil Machine Translation System on Web”, in Kalyansundaram K (ed.) Tamil Internet 2002: Conference Papers, Chennai: Asian Printers, pages 68-73.

Viswanathan, S., Rameshkumar, S. Kumara Shanmugam, B. Arulmozhi. S. & Vijay Shnakar, K.2003. “A Tamil Morphological Analyzer”, in Rajeev Sangal, S.M. Bendre & Udaya Narayana Sigh (eds.), Recent Advances in Natural Language Processing. Mysore: CIIL.

Winston Cruz, S. 2002. Parsing and Generation of Tamil Verbs in GSMorph. M.Phil. dissertation submitted to the University of Hyderabad.

Winston Cruz, S. 2002. Parsing and Generation of Tamil Verbs in GSMorph. M.Phil. dissertation submitted to the University of Hyderabad.

Yagnanarayana, B. et al. 1992. “Text –to-speech system for Indian Languages”, presented at the Workshop on Computer Applications in Indian Languages held at CIIL, Mysore, August 19-21.

Yegnanarayanan B, Rajendran, S, Ramachandaran V.R., Madhukumar, A.S. “Significance of Knowledge Sources for a Text-to-speech System for Indian Languages”, in Sadhana, Academic Proceedings in Engineering Sciences of India.

Yegnanarayanan, B. 1994. “Speech synthesis by Machine”, in B.B. Rajpurohit (ed.) Technology and Languages. Mysore:CIIL.

S. Rajendran, Ph.D.
Department of Linguistics
Tamil University
Thanjavur 613 005
Tamilnadu, India
raj_ushush@yahoo.com